A Laboratory Project on

SECURE AUTHENTICATION SYSTEM

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

BACHELORS OF TECHNOLOGY

In Computer and Communication Engineering

2018-2019

By

AKASH JAIN

179303013

MANIPAL UNIVERSITY
JAIPUR

Under the guidance of

Dr. Naina Narang

Department of Computer and Communication Engineering

School of Computing and Information Technology

Manipal University Jaipur

Jaipur, Rajasthan

1. PROBLEM STATEMENT

The purpose of this project is to design a software program which helps a company to implement a secure authentication system on a Company's server and database. Using this software, a secure method is provided in which an employee can first Sign-up using his mobile number and a unique 10-digit password. Later on, he can Log-In into the Company's system and database using the same Mobile number and Password. Hence, providing a safe environment to access certain information.

2. INTRODUCTION

One of the most important aspects of a database/server authentication focuses on the user and human-to-computer interactions. As a result, user authentication is important to understand or improving your database/server login procedure. A user must prove to the system that they are who they say they are. The ID and key are enough to confirm the user's identity, which will allow the system to authorize the user.

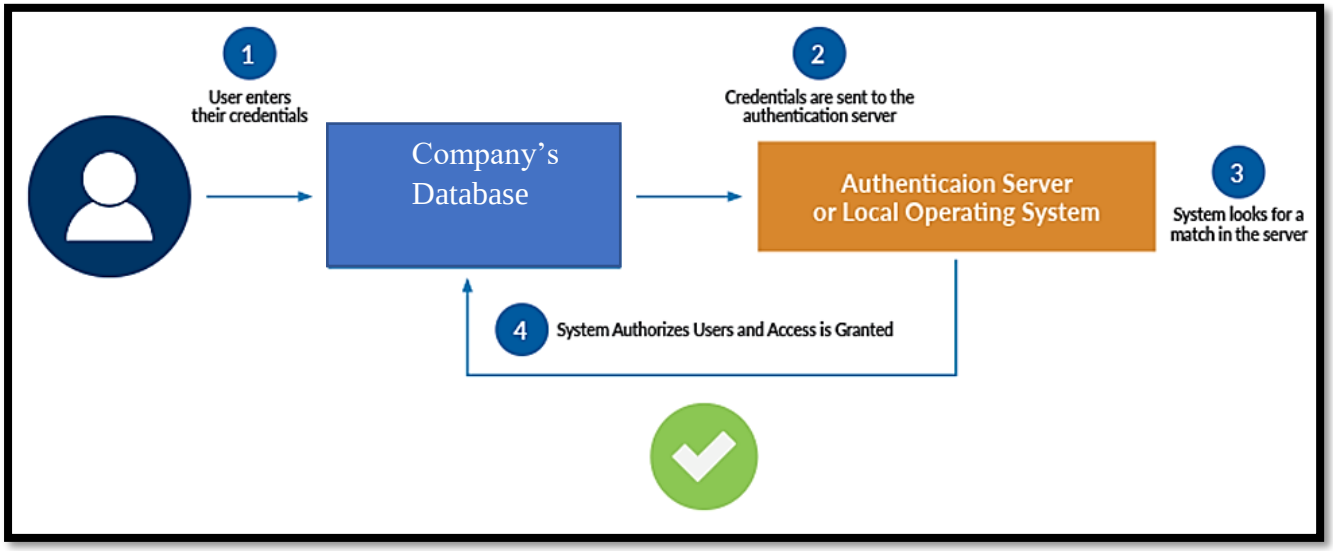
It's important to note that authorization is what dictates what users are able to see and do when they log in. User authentication has basically three tasks:

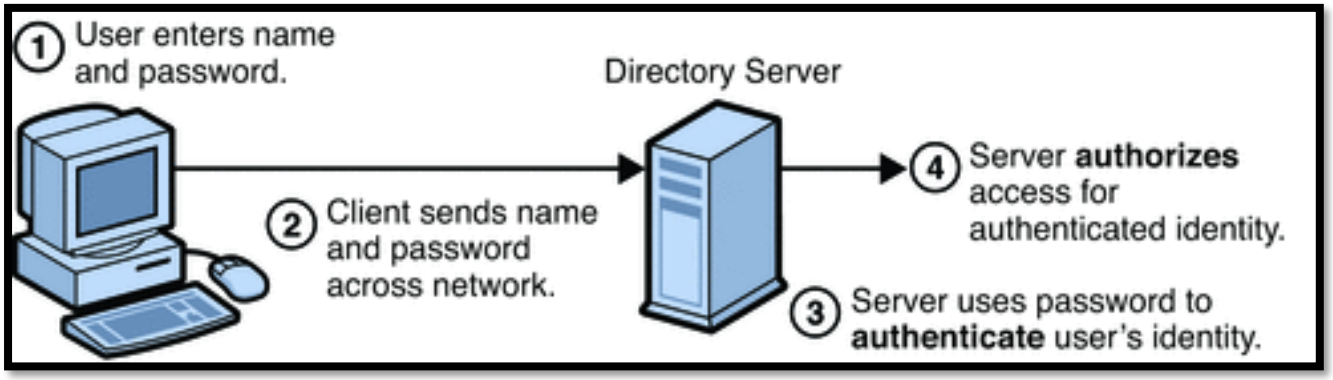
- Manage the connection between the human (user) and the Server/database.
- Verify user's identities.
- Approve (or decline) the authentication so that the system can move to authorizing the user.

User's will input their credentials (here Mobile Number and Password) on the Login form. That information is then sent to the authentication server where the information is compared with all the user credentials on the file (memory). When a match is found, the system will authorize users and grant then access to their accounts.

If a match isn't found, users will be prompted to re-enter their credentials and try again. The entire process takes just a couple of seconds to complete.

3. ABSTRACT GLOBAL LEVEL BLOCK DIAGRAM





4. SPECIFICATIONS

- In the beginning, the user is given three options from which he/she is expected to choose one.
- If the user chooses the Sign-Up option, he/she will be taken to the Sign-up page where the user can create his/her account by providing his/her mobile numbers and a password of having 10 characters. On successfully Signing up, the user will be taken back to the home page.
- While signing up, if the user enters a mobile number which is already in use, then a message will be displayed to tell the user to try and Log in using the mobile number.
- If the user chooses the Log-in option, then he/she will be taken to the Log-in page where he will be prompted to enter the mobile number and password. If a correct match of both the mobile number and password is found, then the user is taken to the Company's database where he is given various options to choose from.
- If the user chooses to Log out, then he/she is taken back to the home page where he/she can again choose to sign-up, login or exit.
- If in the Login page, the user enters the wrong mobile number or password, then a message will be displayed to tell the user that the mobile number and password entered is incorrect and will to prompted back to the login page.
- If the user chooses to Exit, then an Exiting message is displayed.

5. HARDWARE

- An authentication/ directory Server
- A System
- Display Unit

8. SOURCE CODE

```
include emu8086.INC

MYDATA SEGMENT

    A db 0dh,0ah, " 1. Sign Up $";
    B db 0dh,0ah, " 2. Log In $";
    C db 0dh,0ah, " 3. Exit$";
    D db 0dh,0ah, " Enter your choice: $"
    E db 0dh,0ah, " Enter your Phone Number:    $";
    F db 0dh,0ah, " Enter Password (10 Digits): $";
    G db 0dh,0ah, " Sign Up Successful$";
    H db 0dh,0ah, " Sorry this phone number is already taken...try to Log In$"
    I db 0dh,0ah, " Logged In$"
    J db 0dh,0ah, " PhoneNumber or Password doesn't match with our data...try again!$"
    K db 0dh,0ah, "Invalid Input$";
    L db 0dh,0ah, " ****************************************$";
    M db 0dh,0ah, "      SIGN UP$";
    N db 0dh,0ah, "      LOG IN$";
    O db 0dh,0ah, "      EXITING$";
    P db 0dh,0ah, "   PLEASE ENTER A VALID PHONE NUMBER    $"
    Q db 0dh,0ah, "    WELCOME TO WRONGN $"
    R db 0dh,0ah, "1. VIEW YOUR PROFILE $"
    S db 0dh,0ah, "2. VIEW COMPLAINTS $"
    T db 0dh,0ah, "3. VIEW PENDING ORDERS $"
    U db 0dh,0ah, "4. LOGOUT $"
    V db 0dh,0ah, "SUCCESSFULLY LOGGED OUT $"
    Z db 0dh,0ah,"$";
```

```
MYDATA ENDS


MYCODE SEGMENT
START:
; set segment registers:
    MOV AX, MYDATA
    MOV DS, AX

    ; ADD your code here
    MOV CX, -1;
    MOV DI,0000h;

    Back_to_beginning:

        MOV BX,1000h;

        LEA DX,Z;
        MOV AH,9;
        INT 21h;

        LEA DX, A
        MOV AH, 9
        INT 21h         ; Sign Up is printed

        LEA DX, B       ; Log In is printed
        MOV AH, 9
        INT 21h

        LEA DX, C       ; Exit is printed
        MOV AH, 9
        INT 21h
```

```
    LEA DX, D;       ;Enter your choice
    MOV AH,9;
    INT 21h;

    MOV AH, 1      ; Taking choice as input
    INT 21h

    CMP AL,'1';
    JE SignUp;

    CMP AL,'2';
    JE Log_IN;

    CMP AL,'3';
    JE EXIT;

    LEA DX,K       ; Invalid Input
    MOV AH, 9
    INT 21h

    LOOP Back_to_beginning;

SignUP:

    CALL CLEAR
    LEA DX, M;       ;Sign up
    MOV AH,9;
    INT 21h;

    LEA DX, L;       ;****************
    MOV AH,9;
    INT 21h;
```

```
    LEA DX, E;       ;Enter your Phone Number
    MOV AH,9;
    INT 21h;

    MOV BX,1000h;
    MOV CX,10;

Taking_Ph_number:

    MOV AH, 1      ; Taking input
    INT 21h

    CMP AL,30H
    JL  Error

    CMP AL,39H
    JG Error

    MOV [BX],AL;
    INC BX;

    LOOP Taking_Ph_number;

    MOV CX,DI;

    CMP DI,0000h;
    JE INSERT;

    MOV DX,0000h;
    MOV BX,1000h;
    MOV DI,0000h;

Validating_SignUp:
    MOV AX,[BX];
```

```
    SCASW
    JE increment_BX

    MOV BX,1000h;
    MOV DX,0000h;

continue_with_validation:
     CMP DX,5;
     JE Cannot_SignUp;

     CMP DI,CX;
     JE INSERT;

     Jmp Validating_SignUp

increment_BX:

     ADD BX,0002h;
     INC DX;
     jmp continue_with_validation;

Cannot_SignUp:

    LEA DX, L;       ;****************
    MOV AH,9;
    INT 21h;

    LEA DX, H    ; This phone number is already taken...try Log In
    MOV AH, 9
    INT 21h;

    LEA DX, L;       ;****************
    MOV AH,9;
    INT 21h;
```

```
    MOV DI,CX;
    JMP Back_to_beginning;

INSERT:

    MOV DI,CX;

    MOV CX,5;
    MOV BX,1000h;

Inserting:

    MOV AX,[BX];
    MOV ES:[DI],AX;

    ADD BX,0002h;
    ADD DI,0002h;

    LOOP Inserting;

    LEA DX, F;      ;Enter your Password
    MOV AH,9;
    INT 21h;

    MOV CX,10;

Taking_Password:

    MOV AH, 1     ; Taking Input
    INT 21h;
    MOV ES:[DI],AL;
    INC DI;
```

```
    LOOP Taking_password;

    LEA DX, L;      ;****************
    MOV AH,9;
    INT 21h;

    LEA DX, G;      ;Signed Up
    MOV AH,9;
    INT 21h;

    LEA DX, L;      ;****************
    MOV AH,9;
    INT 21h;

    MOV CX,-1;

    JMP Back_to_beginning;

Log_IN:

    CALL CLEAR

    LEA DX, N;      ;Sign up
    MOV AH,9;
    INT 21h;

    LEA DX, L;      ;****************
    MOV AH,9;
    INT 21h;

    LEA DX, E;      ;Enter your Phone Number
    MOV AH,9;
    INT 21h;
```

```
        MOV BX,1000h;
        MOV CX,10;

    Taking_Ph_no:

        MOV AH, 1      ; Taking input
        INT 21h
        CMP AL,30H
        JL  Error1

        CMP AL,39H
        JG Error1

        MOV [BX],AL;
        INC BX;

        LOOP Taking_Ph_no;

        LEA DX,F;       ;Enter your Password
        MOV AH,9;
        INT 21h;

        MOV CX,10;

    Taking_Pass:

        MOV AH,1      ; Taking Input
        INT 21h
        MOV [BX],AL;
        INC BX;

        LOOP Taking_pass;

        MOV CX,DI;
```

```
    MOV DX,0000h;
    MOV BX,1000h;
    MOV DI,0000h;

    CMP DI,CX;
    JE WrongInfo

Validating_LogIn:

    MOV AX,[BX];
    SCASW
    JE increment_BX_

    MOV BX,1000h;
    MOV DX,0000h;

continue_with_validation_dude:
     CMP DX,10;
     JE Loged_In

     CMP DI,CX;
     JE WrongInfo

     Jmp Validating_LogIn

increment_BX_:
        ADD BX,0002h;
        INC DX;
        jmp continue_with_validation_dude;

 WrongInfo:

        LEA DX, L;       ;****************
        MOV AH,9;
```

```
        INT 21h;

        LEA DX, J;      ;Phone Number or Password doesn't match with our data...try
again!
        MOV AH,9;
        INT 21h;

        LEA DX, L;      ;*****************
        MOV AH,9;
        INT 21h;

        MOV DI,CX;
        MOV CX,-1;
        JMP Back_to_beginning

    Loged_In:

        MOV DI,CX;
        MOV CX,-1;

        LEA DX, L;      ;*****************
        MOV AH,9;
        INT 21h;

        LEA DX, I;      ;Logged In
        MOV AH,9;
        INT 21h;

        LEA DX, L;      ;*****************
        MOV AH,9;
        INT 21h;

        call clear;
```

```
    LEA DX, Q;      ;Welcome
    MOV AH,9;
    INT 21h;

    LEA DX, L;      ;****************
    MOV AH,9;
    INT 21h;

    LEA DX, R;      ;Profile
    MOV AH,9;
    INT 21h;

    LEA DX, S;      ;Complaints
    MOV AH,9;
    INT 21h;

    LEA DX, T;      ;Orders
    MOV AH,9;
    INT 21h;

    LEA DX, U;      ;Logout
    MOV AH,9;
    INT 21h;

    LEA DX, D;      ;Choice
    MOV AH,9;
    INT 21h;

    MOV AH, 1     ; Taking input
    INT 21h
    CMP AL,'4'
    JE Log_out;

EXIT:
```

```
        CALL clear

        LEA DX,o;        ;EXIT
        MOV AH,9;
        INT 21h;

        LEA DX, L;       ;****************
        MOV AH,9;
        INT 21h;
        hlt;

    Error:
        LEA DX, P
        MOV AH, 9
        INT 21h

        JMP SIGNUP;

    Error1:
        LEA DX, P
        MOV AH, 9
        INT 21h

        JMP LOG_IN;

    Log_out:
        call clear

        LEA DX, V;       ;Logout
        MOV AH,9;
        INT 21h;

        LEA DX, Z;
        MOV AH,9;
```

```
        INT 21h;

        JMP Back_to_beginning;

    clear:
        MOV AH, 06h      ;
        MOV AL, 00h      ;
        MOV BH, 0Fh      ;
        MOV CX, 0        ; Clear Screen
        MOV DH, 100      ;
        MOV DL, 130      ;
        INT 10h          ;

        MOV DX, 0        ;
        MOV BH, 0        ; Set cursor to (0,0)
        MOV AH, 02h      ;
        INT 10h

        RET              ; Return back to where it was CALLed    ;


ends

end start               ; set entry point and stop the assembler.
```

9. TESTING AND DEBUGGING

The Syntax errors were corrected at the time of assembling and compiling the code.

The logical errors were corrected by running the compiled file in the emulator and then running the program step by step to check the contents of the memory locations and the registers and see if the desired action is being performed. If found to be correct, then the next instruction was executed and the process was repeated.

10. LIMITATIONS

- The length of the password has been fixed as 10. Hence, the user will have to maintain a password of exactly 10 characters.
- The only option available as a user name is that of Mobile Number. The user needs to enter the mobile number compulsorily in order to Sign-up.
- If the user forgets his/her password, then the password cannot be recovered back.

11. BIBLIOGRAPHY

In the completion of the project, some help was taken from external sources though mostly everything was done with what was taught in class. Some external resources from where help was taken are as follows:

- Microprocessor and Microcontroller, D.V. Hall
- emu8086 Documentation